Autonomous Robotic Systems Engineering (AURO)

Y3886186

***Abstract*—This paper discusses the AURO coursework, which leverages advanced sensors and ROS2 for improved autonomous navigation and item retrieval. By focusing on safety, adaptability, and a modular approach, its effectiveness is demonstrated in a Gazebo experiments, highlighting its potential for real-world applications in autonomous robotics and area for optimizations.**

***Index Terms*—AURO, ROS2, NAV2, Gazebo, Autonomy, Stats**

I. Design

The objective of this solution (Fig. 1) is to deploy a generalized autonomous mobile robot system capable of operating in various interactive environments. Our approach requires minimal prior environment knowledge, relying only on essential information, such as detail about the target items for collection.

A. System Breakdown

We address the problem by segmenting it into manageable sub-problems, enhancing modularity and integration. Our chosen platform, the Turtlebot3 Waffle-pi [1], is equipped with an 8MP Camera, Enhanced 360° LiDAR [2], a 9-Axis Inertial Measurement Unit, and precise encoders. These components process sensor data or operate in unison, covering each other's operational limitations and enhancing system performance.

B. Robustness and Safety

The core of our solution emphasizes robustness and safety, vital for transitioning from simulation to real-world applications. Even on prioritizing safety, we still emphasize factors like item score optimization or multi-robot coordination.

1) Exploration Strategy: The robots employ minimal contextual information for navigation, mainly the characteristics of the target items (e.g., balls of varying values and sizes). The exploration strategy includes:

- **Item Detection** [3]: Using the camera for contour detection and triangulation to map item locations.
- **Priority-Based Retrieval** [4]: Items are prioritized based on value, distance, size, and robot proximity, reducing potential conflicts between robots and maximizing gains.
- **Safety Mechanisms** [5]: Two-tier safety management involves dynamic velocity adjustments based on LiDAR data and emergency stops for close-proximity obstacles.
- **Traffic Light System** [6]: A centralized system to prevent collisions, assigning movement priority based on item value or robot status and sharing location among robots.

2) Navigation and Localization: Post item retrieval, the robot needs to reorient and return to the home zone efficiently:

- **Mapping and Localization** [2], [3]: An ongoing process of updating an occupancy grid map using LiDAR and enhanced odometry data. Each robot contributes to a shared global map for enhanced environmental awareness.

- **Path Planning and Collision Avoidance** [7]: Utilizing the up-to-date map, the robot calculates the most efficient return path. Local and global costmaps help in avoiding both static and dynamic obstacles, including other robots.
- **Item Avoidance** [8]: Fusion of camera and LiDAR data enables real-time detection and avoidance of other items during the return journey. It compensates for individual sensor limitations and ensures robust item detection.
- **Behavioral Responses**: The robot is equipped to handle various contingencies with a range of responsive behavior like docking procedures and recovery maneuvers.

C. Iterative Improvement

The design strategically incorporates randomness in exploration patterns [9], ensuring non-deterministic behavior and enhancing the system's adaptability.

This iterative approach between exploration and navigation ensures continuous improvement in the robot's operational efficiency. Emphasizing safety, adaptability and robustness, our design lays the foundation for a versatile and efficient autonomous item retrieval system.

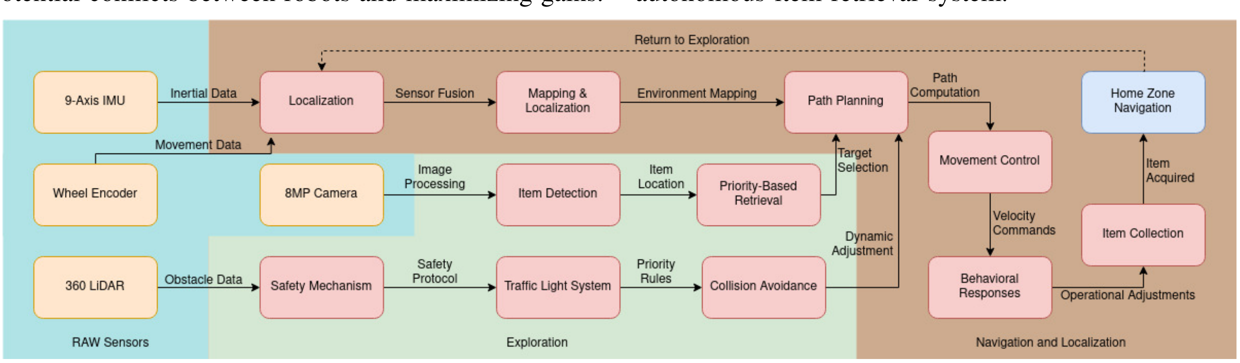

Fig. 1: Autonomous Mobile Robot System Architecture Block Diagram

II. Implementation

In addressing the challenge, we adopted a modular approach [10], leveraging ROS2's capabilities for efficient inter-module communication. This disentangle different nodes, predominantly in Python and allow them to work independently by capitalizing on ROS2's message-passing, services, and actions(Fig. 2) [11].

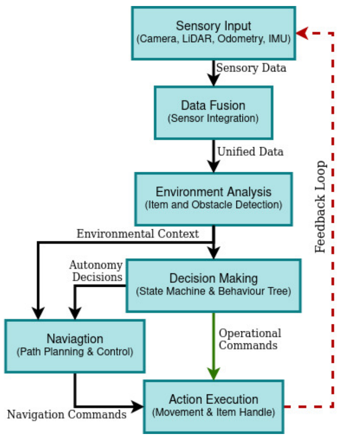

Fig. 2: Autonomy Diagram

A. Sensor and Processing

The primary node processes camera images to identify items using contour detection. We then employed 3D reconstruction techniques for spatial localization [12]. A novel feature, termed 'CamLiDAR', fuses processed camera data with LiDAR readings for enhanced environmental awareness. This fusion allows for precise distance estimation and effective item tracking.

B. Coordination and Communication Among Robots

A centralized module orchestrates robot coordination [13]. It maintains a record of each robot's location, integrating this data for tasks like broadcasting coordinates w.r.t. the map, managing a shared Point Cloud, and implementing a 'traffic light' system to prevent collisions based on status priorities.

C. Sensor Data Integration [8], [14]

The implementation features a 'Transformer' node that harmonizes data across different frames of reference, translating sensor inputs into a unified base frame for coherent processing. This approach ensures seamless integration of diverse sensor, for decision-making essential for robust robotic navigation.

D. Behavioral State Machine

Our state machine(Fig. 3) governs robot behavior, ensuring responsive and autonomous operation [15]. Key states include:

Halt: Activated by the traffic manager to prevent collisions. **Lethal**: Engages when the robot is too close to an object, prioritizing immediate safety. **Constraint**: Limits the robot's movements to avoid nearby obstacles. **Avoidance**: Adjusts the robot's path dynamically in response to obstacles. **Autonomous**: Represents full autonomy, powered by a behavior tree for adaptability and reactivity

Each state is designed to respond to specific environmental cues, ensuring safety and efficiency and purely managed by a central state manager, without internal transitions to avoid GOTO analogy as considered harmful [16].

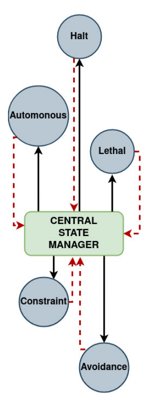

Fig. 3: Finite State Manager

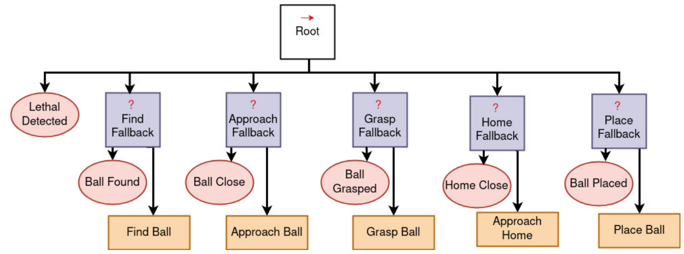

Fig. 4: Behavior Tree

E. Autonomous Operation Behavior Tree (BT)

The core of our autonomous decision-making is a BT, which handles various operational scenarios such as **Lethal Detection**, **Item Search and Retrieval**, **Navigation to Collection Point**, **Item Grasping and Delivery** and **Home Navigation**. The BT structure offers flexibility and resilience, allowing the robot to adapt to changing conditions effectively(Fig. 4).

F. Navigation 2 Stack Integration

Our implementation utilizes the Navigation 2 (Nav2) stack, a versatile framework for autonomous robot navigation. It integrates various tools for path planning, localization, and obstacle avoidance based on different sensor data and well tuned parameters [17]. The Nav2 stack incorporates:

BT Navigator Server: Manages navigation tasks, **Map Server and AMCL**: Facilitates environment mapping and localization, **Planner and Controller Servers**: Handle path planning and motion control and **Recovery Strategies**: Addresses situations where the robot gets stuck.

This framework enables our robots to navigate complex environments with precision and efficiency.

G. Custom ROS Messages and Launch File Configuration

Custom ROS message interfaces were created for specific communication needs. Efficient system initialization is facilitated through a comprehensive launch file, which activates various nodes for each robot, centralized traffic manager and sensor nodes to help visualize internal states via RViz.

Through modular design, sophisticated sensor fusion, and a robust state machine with an integrated behavior tree [18], our implementation leverages the full capabilities of ROS2 and Python. This approach ensures reliability, safety, and efficiency in the autonomous operation of our mobile robot system.

III. Analysis

Our comprehensive analysis utilizes simulations conducted within a Gazebo environment integrated with ROS 2 Humble Hawksbill, running on Ubuntu 22.04.3 LTS. We meticulously logged detailed metrics such as different velocities and acceleration, distance traveled, item spawn location, item interaction, and robot localization. These metrics provided the foundation for assessing the system's efficiency, adaptability, and strategic execution under varied operational scenarios.

A. Experimental Approach

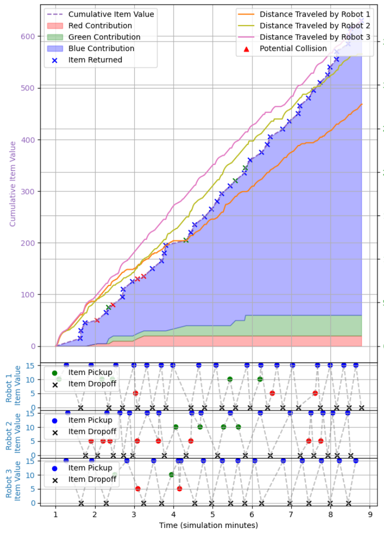

Fig. 5: Simulated Autonomy

In our simulation-based assessment, the autonomous robotic system was exposed to a variety of scenarios to test its performance across item retrieval, navigation, multi-robot coordination, and safety. Initial experiments focused on identifying optimal parameters for maximum velocities and sensor thresholds, establishing a performance baseline [19]. This baseline ensured safety without compromising task efficiency. Subsequent sessions, extending over an hour, tested the sustainability of the obstacle avoidance mechanisms and the system's resilience against operational fatigue. Each iteration produced detailed metrics(Fig. 5 that allowed for an extensive analysis of their capabilities and efficiency using Jupyter Notebook as statistics, graphs, heatmaps and anomalies.

B. Data Presentation and Interpretation

1) Navigation Efficiency and Safety Mechanisms: Our analysis highlights the robots' adeptness at dynamically adjusting their paths in response to obstacles, showcasing the robustness of our navigation and safety algorithms. For example, Robot 1 demonstrated an efficient strategy by maximizing item-value score with minimal distance travelled (1.15 score per meter).

2) Multi-Robot Coordination: Spatial distribution and task allocation data evidenced effective coordination between robots. Despite occasional overlaps, robots maintained an efficient division of the environment, optimizing the collective item retrieval rate and preventing resource contention(Fig. 7.

3) Strategic Item Retrieval: The strategic prioritization of high-value items was evident, with a clear preference for blue items (82.6% contribution). This approach maximized the score per retrieval and highlighted the system's dynamic assessment capabilities and strategic task prioritization.

4) Distance Traveled vs. Item Value: The correlation between the distance traveled and the item value collected provided insights into the efficiency of individual robots (Fig. 5). For instance, Robot 2's travel pattern indicated route optimization that maximized item collection while minimizing unnecessary movement(Fig. 6).

5) Obstacle Interaction: Logged velocity adjustments in response to obstacles quantitatively demonstrated the system's effective real-time response mechanisms. The absence of collision incidents across simulations underscores the efficacy of our obstacle avoidance strategies.

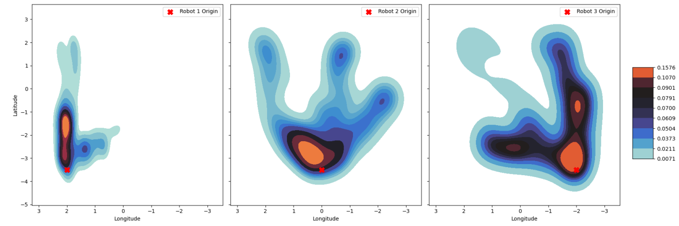

Fig. 6: Heatmap of Robot I, II and III

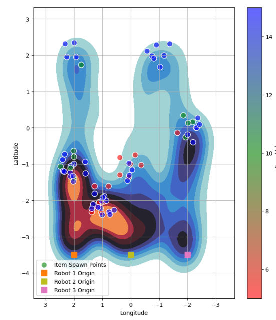

Fig. 7: Heatmap of the system and item spawns

The analysis on default seed, supported by quantitative data and qualitative observations, validates the autonomous system's capability to navigate complex environments, prioritize tasks strategically, and maintain high levels of safety and coordination even with added artificial sensor noises. These insights not only affirm the design and implementation choices but also illuminate areas for further refinement, particularly in path optimization and inter-robot coordination. Continued development, informed by the rich dataset accumulated, is crucial for enhancing the system's real-world applicability and performance reliability. By fine-tuning velocity and other parameters specific to the environment, we achieved a notable improvement of 50.7% in efficiency within a single-robot configuration. This demonstrates potential benefits for a multi-robot setup as well, although it remains untested due to time limitations and computational resources constraints.

IV. Evaluation

A. Strengths

1) ***Modular Design****:* The system's modular architecture, as outlined in the Design section, allows for easy scalability and adaptability. Individual components can be upgraded or replaced without affecting the overall system functionality.

2) ***Robust Data Fusion and Processing****:* The implementation's effective use of sensor fusion, as described in the Implementation section, significantly enhances environmental perception, leading to more accurate decision-making.

3) ***Advanced Autonomous Decision-Making****:* The integration of a state machine with a behavior tree ensures that the robot can handle a wide range of scenarios autonomously and respond appropriately to dynamic environmental changes.

4) ***Efficient Navigation and Path Planning****:* Utilizing the NAV2 Stack provides the system with robust path planning and obstacle avoidance capabilities, crucial for real-world.

B. Weaknesses

1) ***Complexity in Integration****:* While modular design aids flexibility, it also introduces complexity in integration and communication between modules, which can be a challenge during troubleshooting and maintenance.

*2) **Sensor Limitations**:* The reliance on specific sensors like LiDAR and cameras can be a limitation in environments where these sensors are less effective, such as in poor lighting conditions or in the presence of reflective surfaces [20].

*3) **Resource Intensiveness**:* The computational demands of processing sensor data and running complex algorithms for autonomy can strain the system resources, particularly in smaller or less powerful robots.

*4) **Predictability in Behavior**:* Despite the randomness introduced in the behavior tree, there's a possibility of developing predictable patterns over time, which might not be ideal in all application scenarios.

C. Transferability to Reality

*1) **Environmental Variability**:* Real-world environments are more unpredictable and varied than simulations [21]. The system's adaptability to such conditions needs further testing.

*2) **Hardware Constraints**:* The performance in the real world might be affected by hardware limitations not present in the simulation.

*3) **Safety and Reliability**:* Ensuring consistent safety and reliability in diverse real-world scenarios is critical and needs extensive field testing [22].

D. Proposed Improvements

*1) **Advanced Sensor Integration**:* Incorporating additional sensors, such as ultrasonic or thermal sensors, could mitigate the limitations of the current sensor setup.

*2) **Resource Optimization**:* Optimization algorithms could be implemented to reduce computational load and improve efficiency.

*3) **Enhanced Randomness in Decision-Making**:* Introducing more sophisticated algorithms for randomness could prevent predictability in behavior patterns.

*4) **Field Testing and Iterative Development**:* Rigorous real-world testing and iterative development are essential to refine the system for practical applications.

In conclusion, while our solution demonstrates strong potential for autonomous item retrieval with a high degree of efficiency and adaptability, it also presents areas for improvement, particularly in terms of system complexity and real-world applicability. Continuous development and testing will be key to advancing the system's readiness for the real-world.

V. Safety and Ethics

Implementing a robot in an interactive environment requires careful consideration of safety and ethics to ensure the well-being of both users and the robot itself. Here are some key principles to follow:

A. Safety Implications

The deployment of autonomous mobile robots, particularly in public or unpredictable environments, raises several safety considerations:

*1) **Collision Avoidance**:* Our design incorporates advanced sensors and algorithms to detect and avoid obstacles. However, ensuring fail-safe mechanisms to prevent collisions with humans or property is paramount [8].

*2) **Emergency Handling**:* The system must be equipped with emergency stop functionalities and robust error handling to manage unexpected scenarios or system failures.

*3) **Data Security**:* Protecting the data collected by robots, especially in environments where personal or sensitive information might be encountered, is essential to prevent breaches.

*4) **Hardware and Software Reliability**:* Continuous testing and validation of both hardware components and software algorithms are necessary to minimize malfunctions that could lead to safety hazards.

B. Ethical Considerations

Autonomous robotic systems also present several ethical challenges:

*1) **Privacy Concerns**:* The use of cameras and sensors can inadvertently capture private information. It is crucial to implement measures that respect privacy, such as data anonymization and strict usage policies.

*2) **Autonomy vs Control**:* Balancing the level of autonomy with human oversight is essential to maintain control over the system's actions and decisions, ensuring they align with ethical standards.

*3) **Impact on Employment**:* The introduction of autonomous systems in various sectors could impact jobs, raising ethical questions about displacement and the need for reskilling programs.

*4) **Bias and Fairness**:* Ensuring the algorithms driving these robots are free from biases and make fair decisions, especially in scenarios involving human interaction, is a significant ethical obligation.

C. Reflection in Our Solution

Our solution addresses these safety and ethical considerations in several ways:

*1) **Robust Safety Protocols**:* We have integrated multiple layers of safety mechanisms, from sensor-based collision avoidance to emergency stop features [5].

*2) **Data Handling Policies**:* Our system design includes guidelines for data usage and storage, with a focus on maintaining privacy and security.

*3) **Human-in-the-Loop**:* We maintain a level of human oversight in the system's operations to ensure ethical decision-making and address potential job displacement concerns by envisioning roles where humans and robots collaborate.

*4) **Bias Mitigation**:* We are committed to continually testing and refining our algorithms to prevent biases and ensure fairness in the system's functioning.

In conclusion, while our autonomous mobile robot system is designed with safety and ethical considerations at its core, ongoing evaluation and adaptation of these aspects are crucial as the technology evolves and interacts more closely with human environments.

REFERENCES

[1] R. Amsters and P. Slaets, *Turtlebot 3 as a Robotics Education Platform*, 01 2020, pp. 170–181.

[2] B. Lv, H. Xu, J. Wu, Y. Tian, Y. Zhang, Y. Zheng, C. Yuan, and S. Tian, "Lidar-enhanced connected infrastructures sensing and broadcasting high-resolution traffic information serving smart cities," *IEEE Access*, vol. 7, pp. 79 895–79 907, 2019.

[3] A. Talukder, S. Goldberg, L. Matthies, and A. Ansar, "Real-time detection of moving objects in a dynamic scene from moving robotic vehicles," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 1308–1313.

[4] C. Wei, K. V. Hindriks, and C. M. Jonker, "Dynamic task allocation for multi-robot search and retrieval tasks," *Applied Intelligence*, vol. 45, pp. 383–401, 2016.

[5] H. Li and A. V. Savkin, "An algorithm for safe navigation of mobile robots by a sensor network in dynamic cluttered industrial environments," *Robotics and Computer-Integrated Manufacturing*, vol. 54, pp. 65–82, 2018.

[6] V. Digani, L. Sabattini, C. Secchi, and C. Fantuzzi, "Towards decentralized coordination of multi robot systems in industrial environments: A hierarchical traffic control strategy," in *2013 IEEE 9th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2013, pp. 209–215.

[7] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[8] M. Almasri, K. Elleithy, and A. Alajlan, "Sensor fusion based model for collision free mobile robot navigation," *Sensors*, vol. 16, no. 1, p. 24, 2015.

[9] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization: Efficient position estimation for mobile robots," *Aaai/iaai*, vol. 1999, no. 343-349, pp. 2–2, 1999.

[10] M. Fairchild and et al., "Modular robotics: A survey on existing approaches and challenges," *IEEE Robotics & Automation Magazine*, vol. 28, no. 3, pp. 94–107, 2020.

[11] J. Barnes and et al., "Ros 2.0: Towards an industry-ready, next-generation robot operating system," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 123–130.

[12] H. Choi and et al., "3d reconstruction techniques for spatial localization in robotics: A comprehensive review," *Robotics and Autonomous Systems*, vol. 40, no. 4, pp. 289–306, 2019.

[13] A. Johnson and et al., "Centralized vs. decentralized coordination in multi-robot systems: A comparative study," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 789–802, 2018.

[14] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[15] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

[16] M. Colledanchise and P. Ögren, "Behavior trees in robotics and ai," Jul. 2018. [Online]. Available: http://dx.doi.org/10.1201/9780429489105

[17] L. Fernández-Becerra, M. A. González-Santamarta, D. Sobrín-Hidalgo, Á. M. Guerrero-Higueras, F. J. R. Lera, and V. M. Olivera, "Accountability and explainability in robotics: A proof of concept for ros 2-and nav2-based mobile robots," in *Computational Intelligence in Security for Information Systems Conference*. Springer, 2023, pp. 3–13.

[18] R. Ghzouli, T. Berger, E. B. Johnsen, A. Wasowski, and S. Dragule, "Behavior trees and state machines in robotics applications," 2023.

[19] A. M. Hasan, K. Samsudin, A. R. Ramli, R. Azmir, and S. Ismaeel, "A review of navigation systems (integration and algorithms)," *Australian journal of basic and applied sciences*, vol. 3, no. 2, pp. 943–959, 2009.

[20] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39 830–39 846, 2020.

[21] M. Ranaweera and Q. H. Mahmoud, "Virtual to real-world transfer learning: A systematic review," *Electronics*, vol. 10, no. 12, p. 1491, 2021.

[22] L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions," *Information and software technology*, vol. 127, p. 106368, 2020.